APOSTILA DE LaTeX

Programa Especial de Treinamento

Telecomunicações

PET-TELE)))

Universidade Federal Fluminense

Niterói-RJ

2004

Prefácio

Tendo em vista as diretrizes do MEC em Pesquisa, Ensino e Extensão, o Programa Especial de Treinamento do curso de Engenharia de Telecomunicações da Universidade Federal Fluminense (UFF) desenvolveu um projeto de elaboração de apostilas, com o intuito de auxiliar os alunos do curso no aprendizado de temas importantes a sua formação, mas ausentes em quaisquer ementas de disciplinas; E, além disso, servir de material didático para o cursos de capacitação que são dados pelos alunos do programa para os corpos dicente e docente da graduação.

Abaixo segue a lista de apostilas preparadas neste projeto:

HTML Linguagem de programação para hipertextos, principalmente empregada na construção de páginas da Internet (webpages).

LaTeX Sistema de edição de texto largamente utilizado em meios acadêmicos e científicos, bem como por algumas editoras nacionais e internacionais.

LINUX Introdução ao sistema operacional LINUX.

Linguagem C : Linguagem de programação amplamente utilizada em problemas de engenharia e computação.

MATLAB Ambiente de simulação matemática, utilizado em diversas áreas profissionais.

SPICE Ambiente de simulação de circuitos elétricos (analógicos e digitais), utilizado em projeto de circuitos discretos e integrados.

Esta apostila destina-se a introduzir o usuário ao editor de texto LaTeX, explicando como elaborar um documento com os comandos básicos do LaTeX, e não como instalar o programa LaTeX.

Os comandos tratados nesta apostila permitem ao usuário elaborar um bom documento, porém esta apostisla não pretende abordar a todos comandos existentes devido as diferentes áreas onde o LaTeX pode ser usado e a quantidade de funções que são criados a todo momento (veja secção2.1.1). Para maiores informações e um estudo mais aprofundado ao LaTeX consulte as referências bibliográficas no fim do documento.

Sumário

Capítulo 1

O que é o LaTeX?

Antes de saber o que é o LaTeX é preciso conhecer o TEX. O TEX é um programa criado por Donald Knuth na década de 70 com a finalidade de aumentar a qualidade de impressão com base nas impressoras da época e é utilizado para processar textos e fórmulas matemáticas.

LaTeX é um programa que reúne comandos que utilizam o TEX como base de processamento e foi criado por Leslie Lamport na década de 80 com o objetivo de facilitar o uso do TEX através de comandos para diferentes funções.

É um editor de textos especialmente voltado para a área matemática contendo comandos para montar as mais diversas fórmulas. Gera textos de alta qualidade tipográfica (espaçamento entre palavras, combinação de letras etc...). Além de ser muito bom para fazer textos grandes como livros.

O LaTeX é um processador baseado no estilo lógico. Os programas de processamento de texto podem ser divididos em duas categorias:

Estilo visual Nestes processadores de texto, existe um menu na tela apresentando os recursos que podem ser usados, sendo selecionados através do mouse. E o texto que você digita aparece na tela da mesma forma que vai ser impresso. Isso é conhecido como WHAT-YOU-SEE-IS-WHAT-YOU-GET (WYSIWYG). Ex: Word.

Estilo lógico Nesta categoria o processamento é feito em duas etapas distintas.

- O texto a ser impresso e os comandos de formatação são escritos em um arquivo fonte com o uso de um editor, (isso não impede que também haja um menu na tela onde os comandos podem ser selecionados, isto é apenas um adicional oferecidos por fabricantes para facilitar a digitação).
- Em seguida este arquivo é compilado e gera uma arquivo de saída que pode ser visualizado. Ex: HTML.

Os comandos LaTeX foram criados com base em diversos estudos sobre diagramação. Isto foi importante para fazer com que o LaTeX entenda o que o autor deseja fazer, por exemplo, como diferenciar um texto matemático de uma citação de fala. Segundo os estudos, existem formas que permitem tornar o texto muito mais claro. O tamanho deve ter um limite ideal para facilitar a leitura assim com o da fonte, o espaçamento das linhas, letras e palavras também têm uma medida ideal.

Geralmente, quando o autor está trabalhando com a um processador visual, ele comete muitos erros por não conseguir conciliar uma boa estética com uma estrura lógica e bem

compreensível. Utilizando o LaTeX, que é um processador lógico, o autor se preocupará mais com o conteúdo. Dessa forma, seu texto não terá apenas uma boa estética e estrutura coerente, mas também com um bom conteúdo. Além disso, com a troca de apenas um comando, algo pode ser mudado futuramente com facilidade, o que deixa o documento muito mais flexível.

Vamos ver o que acontece tecnicamente:

O autor escreve seu documento usando os comandos do LaTeX. O LaTeX entende o que o autor quis dizer e transforma os comandos digitados em uma linguagem inteligível pelo TEX, ou seja, o texto escrito em linguagem LaTeX é processado por um compilador seguindo as regras dessa linguagem. Isto é, transforma um arquivo *.tex (que possui o código *tex*) em um arquivo *.dvi (*device independent*). Este último é o documento pronto para ser visualizado.

Claro que não se pode esquecer que seu computador deve ter a biblioteca do LaTeX para fazer isso. Alguns dos sistemas TEX, atuais, disponíveis no mercado são: fptex, pctex, miktex (Windows) e tetex (Linux, embora provavelmente já o tenha instalado).

Esse arquivos *.dvi podem ser lidos independentemente da versão do editor usado, logo pode ser lido em qualquer sistema, contanto que ele tenha um programa específico para lê-lo.

Outra vantagem do LaTeX é sua estabilidade, ou seja, a probabilidade de se encontrar um *bug* nele e mínima e justamente por ser *free software* seu sistema é aberto, o que permite que qualquer um corrija possíveis *bugs* ou que possa adaptá-lo às suas necesidades. A cada momento surgem novos pacotes com funções criadas por usuários espalhados por todo o mundo.

Capítulo 2

Documento

2.1 Estrutura

A linguagem LaTeX funciona a base de comandos que são iniciados com \ que é um marcador (*tags*, de *Tag languages*).

Os comandos são escritos nas formas `\comando` ou `\begin{comando}...\end{comando}`. Quando vem escrito nesta última forma, ele é chamado de **ambiente**.

O texto de cada tipo de documento começa com
`\begin{document}` e termina com `\end{document}`. Tudo o que vem antes disso é considerado o **preâmbulo** e tudo o que vem depois de `\end{document}` é ignorado.

É no preâmbulo que são colocadas todas as imformações referentes às principais características que terá seu documento. Começa com `\documentstyle{estilo}` no caso do LaTeX2.09 e com `\documentclass{estilo}` no caso do LaTeX 2_ε segunda edição.

No lugar de estilo é colocado o nome de um dos estilos pré-definidos, como:

article Textos pequenos;

report Relatórios;

book Livros, apostilas;

letters Cartas.

Obs: Os estilos não são apenas estes. Geralmente congressos, universidades e outros meios disponibilizam outros estilos de formatação para apresentação de trabalhos. Isso mostra uma das vantagens do LaTeX, que é a flexibilidade para se criar novas formatações que atendem a diferentes nescessidades.

Podendo, também, ser selecionadas algumas opcões dentro do estilo escolhido, como:

- **Tamanho**: Padrão da letra: 11pt ou 12pt(pontos), o último é usado com mais frequência;

- **twoside**: Que imprime em ambos os lados da página;

- **oneside**: Imprime em um só lado da página;

- **twocolumn**: Produz o texto disposto em duas colunas na página;

- **onecolumn**: Produz o texto disposto em uma coluna;
- **landscape**: Produz uma página na forma de paisagem;
- **leqno**: Isto faz com que a numeração das fórmulas sejam colocadas a esquerda em vez de a direita;
- **fleqn**: Faz com que a fórmula fique localizada na margem esquerda em vez de estar centralizada;
- **openright**: Faz com que os capítulos sejam iniciados apenas nas páginas ímpares;
- **openany**: Permite que os capítulos sejam iniciados nas páginas ímpar ou par.
- **Tamanho da folha**: Pode ser a4, letterpaper etc...

Obs: Existe o comando `\twocolumn` que inicia uma nova página produzindo duas colunas nela a partir do ponto onde foi colocado no meio do texto. Caso isto não tenha sido declarado no preâmbulo o comando `\onecolumn` faz o inverso. Para reverter à mesma maneira que está no preâmbulo use o comando oposto.

Essas opções são colocadas entre colchetes sem espaço entre as palavras e com vírgula.

Ex:

```
\documentstyle[twocolumn,12pt,a4]{article}
\usepackage{pacote}
\begin{document}
.
.
.
\end{document}
```

2.1.1 Pacotes

Pode-se definir pacotes como um conjunto de arquivos que implementam uma determinada característica adicional para os documentos escritos em LaTeX.

Quando o usuário quiser montar um documento um pouco mais elaborado, perceberá que os comandos básicos que o LaTeX contém não são suficientes, sendo necessário, o uso de algo que aumente a sua capacidade.

Alguns pacotes já vem como distribuição básica do LaTeX, outros podem ser encontrados separadamente, veja a referência [4], pois a todo momento novos pacotes são criados por usuários em todo o mundo.

Estes pacotes são inseridos no preâmbulo usando o comando
`\usepackage[opcional]{pacote}`

Ao longo de toda a apostila será apresentado pacotes com diferentes funções. Abaixo, segue a lista com uma breve descrição de alguns deles. O funcionamento de cada um será explicado posteriormente.

Os principais são:

graphicx Para inserir gráficos. Veja seção 5.4;

color Para usar cor no seu texto. Veja secção 6.5;

babel Para traduzir nomes que aparecem em inglês na estrutura do documento. Ex: *chapter*, *section*, *tableofcontents*, etc. Neste caso para que estas palavras sejam traduzidas para o português use o comando opcional [brazil].

fontenc Permite que o LaTeX compreenda a acentuação feita direto pelo teclado. É usado com o opcional [T1].

amsfonts Define alguns estilos de letras para o ambiente matemático;

fancyhdr Para fazer cabeçalhos personalizados.Veja secção 6.1.

Obs: Nem todos os pacotes são compatíveis com qualquer versão do LaTeX. Os criadores sempre quando criam novos pacotes, tentam deixá-los compatíveis com qualquer versão do LaTeX, porém alguas vezes isso não é possível.

É muito simples saber se seu sistema possui determinados pacotes, ou instalá-los, basta consultar o manual dele. Caso o usuário esteja escrevendo algum documento e precise mudar constantemente de computador, mas não sabe se o sistema deste possui o pacote que seu documento precisa, é só copiar todos os arquivos do pacote e deixar no mesmo diretório em que seu documento está. Quando o LaTeX está compilando o documento, o arquivo do pacote será procurado no caminho padrão do sistema ou no prórpio diretório do seu documento.

Por exemplo, imagine que você esteja escrevendo um trabalho e que ele deva seguir um determinado modelo que o congresso exija. Provavelmente, seu sistema não terá instalado o pacote que faz isto. O que fazer então? É só pegar os arquivos deste pacote e deixar no mesmo diretório do arquivo do trabalho. Quando compilar, o LaTeX lerá os arquivos deste pacote e gerará o documento no formato desejado. E se for preciso mudar o sistema operacional, não haverá problema algum, basta andar sempre com estes arquivos (que são pequenos, poucos *Kilobytes*) junto com arquivos de seus documentos.

Cada pacote possui um manual com os comandos e suas funções. Alguns dos pacotes descritos nesta apostila foram explicados de acordo com o manual deles que estavam disponíveis no próprio sistema. Lembrando, basta um pouco de curiosidade para descobrir a quantidade de recursos oferecidos pelo LaTeX. Basta ler o manual do sistema e descobrir como trabalhar com pacotes. Há uma infinidade deles com as mais diferentes funções.

2.2 Texto

2.2.1 Acentuação

Quando o pacote *fontenc* não tiver sido declarado, a acentuação no LaTeX é feita com comandos da seguinte maneira:

- `\c{c}` – ç
- `` \`{e} `` – è
- `\'{e}` – é
- `\^{e}` – ê

- \~{e} – ẽ
- \"{q} – q̈

Outros tipos de acentos estão no apêndice C.

2.2.2 Sentenças e Parágrafos

Estamos acostumados a visualizar o espaçamento entre palavras de acordo com o número de vezes em que apertamos a tecla de espaçamento. Já no LaTeX, isso não importa dado que sempre será contado apenas um independentemente da quantidade de espaços inseridos.

O primeiro parágrafo será iniciado sem indentação, como pode ser observado logo acima. O comando `\indent` adiciona uma largura igual ao tamanho da indentação do paragrafo normal e o comando `\noindent` retira a indentação do local onde ela deveria aparecer. Estes comando funciona somente para alguns estilos.

Para ser fazer o próximo parágrafo basta pular uma linha ou utilizar o comando `\par` no lugar em que será iniciado o novo parágrafo.

Para passar para a linha abaixo da qual se está digitando coloque \\ e a linha será quebrada neste ponto. Ex:

```
observe que esta linha está\\
quebrada após o\\
uso das duas barras.
```

observe que esta linha está
quebrada após o
uso das duas barras.

Usando o comando `\linebreak` a linha é quebrada e a parte anterior ao comando fica justificada.

Ex:

```
Esta linha  está \linebreak  justificada.
```

Esta linha está
justificada.

Quando é usado o comando `\newpage` o texto passa para a próxima página. O comando `\pagebreak[num]` força a quebra da página, onde o argumento opcional [num] é um valor inteiro de 1 a 4 que definem a prioridade do funcionamento do comando, onde 4 é o maior valor. O comando `\nopagebreak[num]` faz o inverso de *pagebreak*, ou seja, impede a página de ser quebrada no local onde o comando foi colocado.

Obs: Os comandos `\\*` e `\\[medida]` tem a mesma função do comando `\\`. A diferença está no fato de que `\\*` impede que o texto mude de página na quebra daquela linha e que `\\[medida]` quebra a linha,porém acrescentando o espaço que está determinado entre colchetes. Veja as unidades de medida na secção 6.3

2.2.3 Alinhamento

Usando o ambiente *center* o texto poderá ser centralizado.

Com o ambiente *flushleft* o texto é alinhado a esquerda.

E com o ambiente *flushright* o texto é alinhado a direita.

`\begin{center}`
`  texto`
`\end{center}`
	texto (centralizado)

`\begin{flushleft}`
`texto`
`\end{flushleft}`
	texto

`\begin{flushright}`
`texto`
`\end{flushright}`
	texto (à direita)

Obs: As linhas são apenas para facilitar a visualisação.

2.2.4 Símbolos especiais

O LATEX possui 10 caracteres especiais com os quais são digitados comandos:
`~,^,\ ,#, $, %, & , _ , { , }`

Para que alguns desses apareçam no texto, ponha \ na frente. Veja a tabela de símbolos no apêndice.

2.2.5 Traços (-)

Podem ser feitos três diferentes tamanhos de traços se digitados uma, duas ou três vezes (-)
Ex:

-	com `-`
–	com `--`
—	com `---`

2.2.6 Estilos de letras e Tamanhos

As palavas podem ser colocadas em:

`\textbf{negrito}`	**negrito**
`\textsf{sans serif}`	sans serif
`\textsl{slanted}`	*slanted*
`\textsc{small caps}`	SMALL CAPS
`\texttt{letra de máquina}`	`letra de máquina`
`\textrm{romano}`	romano

Os comandos `\bf`, `\it`, `\sf`, etc..., também podem ser usados e fazem parte do LaTeX2.09.

O texto inteiro também pode ter o tipo de letra diferente do romano, que é o padrão. Basta usar os comandos:

- `\sffamily` - Para colocar o texto no tipo sans serif
- `\ttfamily` - Para colocar o texto no tipo `letra de máquina`
- `\rmfamily` - Para colocar o texto no romano

A partir do ponto onde estes comandos forem colocados é que será mudado o estilo da letra.

Obs: É possível que seu sistema possua outras fontes, basta consultar os pacotes que estão instalados em seu computador.

E os tamanhos podem ser:

{\tiny{tamanho}}	→	tamanho
{\scriptsize{tamanho}}	→	tamanho
{\footnotesize{tamanho}}	→	tamanho
{\small{tamanho}}	→	tamanho
{\normalsize{tamanho}}	→	tamanho
{\large{tamanho}}	→	tamanho
{\Large{tamanho}}	→	tamanho
{\LARGE{tamanho}}	→	tamanho
{\huge{tamanho}}	→	tamanho
{\Huge{tamanho}}	→	tamanho

Usando os comandos para fazer ambientes pode-se mudar o tamanho e a forma das palavras em vários paragráfos. Ex:\begin{huge}...\end{huge}.

2.2.7 Prevenindo a quebra de palavras

Pode acontecer quebra de alguma palavra na troca de linha ou página de forma errada, pois o LaTeX não utiliza a regra de hifenização das palavras em português. Para evitar isso use \mbox{palavra}

Há também outra maneira: colocar no preâmbulo o comando \hyphenation{pa-la-vra} com a palavra dividida em sílabas da forma correta.

Mas é recomendável que só se faça este tipo de coisa quando tiver sido terminado o texto, pois conforme o texto vai sendo digitado a disposição deste na tela pode mudar.

2.2.8 Notas

As notas de pé de página são produzidas com o comando \footnote{texto}

Ex: Esta frase tem uma nota no fim da página [1].

Há também uma maneira de colocar as notas na margem da página. Basta colocar o comando \marginpar{texto}. A nota ficará na altura da linha de texto em que foi colocada.

Ex: Esta senteça tem uma nota\marginpar{esta ...página} na margem.

esta é a nota na margem da página

2.2.9 Fórmulas

Nos textos podem ser inseridas fórmulas com os seguintes comandos:

- \(fórmula\)
- $fórmula$

[1] esta é a nota do pé da página

- `\begin{math}fórmula\end{math}`

A maneira como se faz as fórmulas e o resultado do uso de cada comando será vista mais adiante no capítulo 3.

2.2.10 Comentário

Utilizando o caracter % no início de uma linha do código fonte de documento o LaTeX ignora o que está escrito nela na hora de compilar. Esse caracter é considerado um marcador de comentário.

Ex:

```
% este é o comentário no código fonte
```

2.2.11 Título do documento

É feito com os seguintes comandos:

```
\title{título}
\author{autor}
\date{data} -> este é opcional, se não usar é
                     colocada a data atual
\begin{document}
\maketitle
```

Caso tenha mais que um autor pode ser colocado da seguinte forma:
`\author{primeiro \and segundo}`

Outra maneira de se fazer o título é usando o ambiente *titlepage* que é colocado após o ambiente *document*. Neste ambiente há a liberdade de montar a capa a maneira desejada, sem precisar usar os comandos citados acima.

2.2.12 Resumo

Um parágrafo com o título de resumo pode ser feito usando o ambiente *abstract*. Geralmente ele é colocado na página de título ou página separada, para que o autor dê uma breve explicação sobre o documento. Ele só é válido nos estilos *report* e *article*.

2.2.13 Secções

Em textos um pouco mais longos podem haver varias secções. Então, o LaTeX contém alguns comandos para dividir seu texto deixando-o mais organizado e com estrutura coerente. São eles:

```
\part{parte}
\chapter{capítulo}
\section{secção}
\subsection{sub-secção}
```

```
\subsubsection{sub-sub-secção}
\paragraph{parágrafo}
\subparagraph{subparágrafo}
```

Todas essa partes e sub-partes são numeradas seguindo uma estrutura lógica. Mas colocando um asterisco após o comando não ocorre a numeração
Ex: `\section*{secção}`.

Obs: Os comandos *part* e *chapter* só podem ser usados com os estilos **report** ou **book**.

2.2.14 Identação

Para facilitar a leitura de algumas sentenças é necessário que as enfatize, o que é feito através do ambiente:

```
\begin{quote}
    sentença que está enfatizada
\end{quote}
```

Ex: Esta é uma das célebres frases de um dos maiores cientistas do século passado: Albert Einstein.

> "O importante é não parar de questionar. A curiosidade tem sua própria razão para existir. Uma pessoa não pode deixar de se sentir reverente ao contemplar os mistérios da eternidade, da vida, da maravilhosa estrutura da realidade. Basta que a pessoa tente apenas compreender um pouco mais desse mistério a cada dia. Nunca perca uma sagrada curiosidade".

Também pode ser usado o comando:

```
\begin{quotation}
    sentença que está enfatizada
\end{quotation}
```

A diferença do anterior deste é que o último permite que seja enfatizado vários paragráfos, devido a sua identação.

2.2.15 Listas

Na hora de se construir itens é que pode-se perceber uma das grandes facilidades proporcionadas pelo LaTeX, pois há comandos próprios pra isso.

Para listas:

- Somente com marcação, usa-se:

```
\begin{itemize}
    \item primeiro item
    \item segundo item
    \item terceiro item
\end{itemize}
```

– primeiro item

– segundo item

– terceiro item

São permitidos colocar alguns símbolos para fazer um marcador personalizado no item, como: $\heartsuit$, $\diamondsuit$, $\spadesuit$, $\sharp$, § , † .Eles são colocados assim: `\item[comando do simbolo]` texto. Veja como fazer estes símbolos no apêndice B.

```
\begin{itemize}
    \item[$\clubsuit$]item com marca-
    dor personalizado.
\end{itemize}
```

$\clubsuit$ item com marcador personalizado.

- Listas numeradas usa-se:

```
\begin{enumerate}
    \item primeiro item
    \item segundo item
    \item terceiro item
\end{enumerate}
```

1. primeiro item
2. segundo item
3. terceiro item

- Para listas com descrição é usado:

```
\begin{description}
\item[item] descrição deste
\item[item] descrição deste
\end{description}
```

item descrição deste

item descrição deste

2.2.16 Versos

Para fazer a construção de um verso basta usar o ambiente:

```
\begin{verse}
  verso
\end{verse}
```

Ex: Esta é a parte de uma poesia de Gonçalves Dias.

```
\begin{verse}\index{verse}
... \\
Nosso céu tem mais estrelas\\
Nossas várzeas têm mais flores\\
Nossos bosques têm mais vida\\
Nossa vida mais amores\\
...
\end{verse}
```

...
Nosso céu tem mais es-
trelas
Nossas várzeas têm mais
flores
Nossos bosques têm mais
vida
Nossa vida mais amores
...

Note que deve ser usado \\ para pular linhas. Note também a forma como acontece a quebra da frase que não cabe na mesma linha.

2.2.17 Símbolos

Alguns símbolos e caracteres de língua estrangeira podem ser gerados com o LaTeX.

Ex:
`\c{o}` = ǫ
`\S` = §
`\copyright` = ©

Veja mais símbolos nos apêndices.

2.2.18 Textos pré-formatados

O LaTeX também permite que seja digitado algo da mesma forma que deverá apareçer na tela ou trechos de texto que possuem muitos caracteres. Isso é feito com o ambiente *verbatim*. Dentro desse ambiente pode ser digitado qualquer coisa, até mesmo os espaços são colocados da mesma forma. Isto é muito útil na hora de digitar textos na forma de uma linguagem de programação por possuir muitos caracteres, que também são usados para fazer comando em LaTeX.

Usando o ambiente, seu texto ficará evidenciado em uma linha a parte, mas para que ele continue na mesma linha em que está sendo digitado, use o comando `\verb= seu texto =` , onde este sinal de igual pode ser substituido por qualquer coisa desde de que não seja espaço, asterisco ou letras.

Ex: texto `\verb+{|@#$%+` texto, você verá: texto `{|@#$%` texto.

Se for usado *verbatim* ou `\verb` seguido de um asterisco, em vez dos espaços em branco será colocado o símbolo: ␣.
Ex: `\verb*=a b c d=` a␣b␣c␣d

É importante lembrar que dentro do ambiente *verbatim* o comando que você colocar não será considerado.

Capítulo 3

O ambiente matemático

3.1 Onde fica a fórmula ?

As fórmulas matemáticas podem ser digitadas tanto no meio de um texto ou em destaque:

- No meio do texto:

```
Segundo a equação:
$a^{2}= b^{2}+c^{2}$
 concluímos que...
```

Segundo a equação: $a^2 = b^2 + c^2$ concluímos que...

Tem que ser usado $...$ para que a equação apareça no meio do texto. Além disso, podem ser usados:

```
\( fórmula \) ou \begin{math} fórmula \end{math}
```

- Em destaque:

```
Segundo a equação:
\begin{equation}
a^{2}= b^{2}+c^{2}
\end{equation}
podemos concluir que...
```

Segundo a equação:

$$a^2 = b^2 + c^2 \tag{3.1}$$

podemos concluir que...

Outra maneira para fazer a equação apareçer em destaque é usando os ambiente:

```
\[ fórmula \] ou \begin{displaymath} fórmula \end{displaymath}
```

Neste modo a equação é numerada automaticamente de acordo com a secção. para que isso não aconteça use o comando `\nonumber` dentro do ambiente.

3.2 Construindo fórmulas

3.2.1 Sobescritos e Sobrescritos

Sobrescrito – É feito usando: `b^{e}` onde *b* é a base e *e* o expoente.
Ex: `2^{5}` → 2^5 Sobescritos – É feito usando: `b_{i}` onde *b* é a base e *i* o índice.
Ex: `2_{5}` → 2_5

3.2.2 Frações

Podem ser feita usando:

- /
  Ex: `(a+b)/2`→ $(a + b)/2$
- `\frac{numerador}{denominador}`
  Ex:`\frac{a+b}{2}`→ $\frac{a+b}{2}$

3.2.3 Raízes

São feitas usando: `\sqrt[]{}` Ex:`\sqrt[3]{8}` → $\sqrt[3]{8}$
Se for omitido o termo [ ] automaticamente a raíz será quadrada.

3.2.4 Símbolos matemáticos

O LaTeX possui vários símbolos para montar fórmulas como integrais, somatórios, letras especiais etc...
Ex:
`\int`→ $\int$
`\exists`→ $\exists$
`\infty`→ ∞

Veja mais no apêndice B.

3.2.5 Funções

O LaTeX também possui símbolos de funções.
Ex:
`\log10` → $\log 10$
`\sin60` → $\sin 60$

Veja como usar sobescritos em algumas funções como limite e somatório.

```
\[
\lim_{x\rightarrow\infty} \frac{x^{3}}{e^{x}}
\]
```

$$\lim_{x\rightarrow\infty} \frac{x^{3}}{e^{x}}$$

```
\[
\sum_{n=0}^{\infty} a_{n}^{2}
\]
```

$$\sum_{n=0}^{\infty} a_{n}^{2}$$

```
$\sum_{n=0}^{\infty} a_{n}^{2}$
```

$\sum_{n=0}^{\infty} a_{n}^{2}$

Os intervalos são colocados da mesma forma que se põe sobre e sobescritos. E repare na diferença da disposição dos intervalos quando é usado fórmula em destaque e no meio do texto.

Ex:

```
\[
f(t)= \frac{A}{2} + \frac{jA}{2 \pi }
\sum_{\stackrel{-\infty}{n
\neq 0}}^{\infty} \frac{1}{n} \, e^{jn2\pi t}
\]
```

$$f(t)= \frac{A}{2} + \frac{jA}{2 \pi } \sum_{\stackrel{-\infty}{n \neq 0}}^{\infty} \frac{1}{n} \, e^{jn2\pi t}$$

No apêndice B há uma lista com as funções.

3.2.6 *Array*

É um ambiente que separa os itens em linhas e colunas. A posição do item em relação à sua coluna é especificado por uma simples letra (c-centro, r- direita, l-esquerda). As linhas são separadas usando \\ e as colunas com o simbolo &. Após a última coluna não deve ser colocado &. Também não deve se esquecer de colocar algo para indicar que é um ambiente matemático.

Ex 1:

```
\begin{equation}
      \begin{array}{lr}
        x & y \\
        z & w \\
      \end{array}
   \end{equation}
```

$$\begin{array}{lr} x & y \\ z & w \\ \end{array} \tag{3.2}$$

Ex 2:

```
\begin{equation}
\int_{-L}^{L} sen \frac{m \pi x}{2}\,sen \frac{n \pi x}{2}\,dx =
\left \{
  \begin{array}{cc}
  0, & m \neq n  \\
  1, & m = n \\
\end{array}
\right.
\end{equation}
```

$$\int_{-L}^{L} sen \frac{m \pi x}{2}\,sen \frac{n \pi x}{2}\,dx = \left \{ \begin{array}{cc} 0, & m \neq n \\ 1, & m = n \end{array} \right. \tag{3.3}$$

Através dos seguintes comandos opcionais:

t A linha do topo do *array* se mantém na mesma altura da linha do texto anterior e posterior.

b A linha debaixo do *array* se mantém na mesma altura da linha do texto anterior e posterior.

Vamos ver um exemplo para que fique mais claro.

Deve ser lembrado que estes são comandos opcionais, logo devem ser colocados entre colchetes.

Ex 3:

```
\[
      x -
       \begin{array}[t]{c}
                a\\
                b
       \end{array}
               -
       \begin{array}[b]{c}
                x\\
                y
       \end{array}
 \]
```

$$x - \begin{array}[t]{c} a\\ b \end{array} - \begin{array}[b]{c} x\\ y \end{array}$$

3.2.7 Delimitadores

São símbolos que limitam a expressão como parentêses, chaves e colchetes. É usado `\left` para a limitar parte esquerda e `\right` para a parte direita. Fazendo a combinação destes símbolos com o ambiente *array* podem ser construídas as matrizes.

Ex:

```
\[
        \left(
        \begin{array}{c}
           x\\
           y
        \end{array}
        \right)
\]
```

$$\left(\begin{array}{c} x\\ y \end{array} \right)$$

Ex:

```
\begin{equation}
   \frac{d}{dt}\left ( \begin{array}{c}
     u \\ v
   \end{array} \right) = \left ( \begin{array}{cc}
     -1 & -2 \\
      0 & -1 \\
   \end{array} \right ) \left ( \begin{array}{c}
     u \\ v
   \end{array} \right)
\end{equation}
```

$$\frac{d}{dt}\left(\begin{array}{c} u \\ v \end{array} \right) = \left(\begin{array}{cc} -1 & -2 \\ 0 & -1 \end{array} \right) \left(\begin{array}{c} u \\ v \end{array} \right) \tag{3.4}$$

No lugar de (,) ,no exemplo acima, poderiamos ter colocado: { , } [,] , |. Quando um ponto (.) for usado ápós *left* ou *right* não aparecerá delimitador, observe o uso disto na equação 3.3

3.2.8 Fórmulas em várias linhas

É uma combinação do ambiente *array* com o ambiente matemático de equação, chamado de *equanarray*. Também são usados & e \\ com a opção de usar `\nonumber` para não numerar as fórmulas e de deixar de usar os argumentos c,l,r.

Ex 1:

```
\begin{eqnarray}
x & = & m  +  n  +  p \\
y & = & z  +  w  +  u \nonumber \\
x &   & p  +  n
\end{eqnarray}
```

$$\begin{array}{rcll} x & = & m+n+p & (3.5) \\ y & = & z+w+u & \\ x & & p+n & (3.6) \end{array}$$

Repare no espaço em branco na última linha entre os & e o resultado disso. E também no uso de `\nonumber`.
Para que nenhuma equação seja numerada é só usar o ambiente `{eqnarray*}`.

Ex 2:

```
\begin{eqnarray*}

\int_{-L}^{L} f(x)\,dx &=& a_{0} \int_{-L}^{L}dx +
                           \sum_{m=1}^{\infty} a_{m}
                           \int_{-L}^{L} cos \frac{m \pi x}{2}\, dx \\
                       &+& \sum_{m=1}^{\infty} b_{m}
                            \int_{-L}^{L} sen \frac{m \pi x}{2}\, dx
\end{eqnarray*}
```

$$\int_{-L}^{L} f(x)\,dx = a_0 \int_{-L}^{L} dx + \sum_{m=1}^{\infty} a_m \int_{-L}^{L} cos\frac{m\pi x}{2}\,dx$$
$$+ \sum_{m=1}^{\infty} b_m \int_{-L}^{L} sen\frac{m\pi x}{2}\,dx$$

Como visto anteriormente, cada equação recebe uma diferente referência. Porém, se o usuário desejar usar a mesma referência para todas as equações é só utilizar o pacote chamado *subeqnarray*.

No preâmbulo deve ser colocado:

```
\usepackage{subeqnarray}
```

E lugar do ambiente *eqnarray* use *subeqnarray*.

Ex:

```
\begin{subeqnarray}\label{eq:sub}
\slabel{sub1} a^2& =& b^2 + c^2\\
\slabel{sub2} a &=& b - 5
\end{subeqnarray}
```

Faz:

$$a^2 = b^2 + c^2 \quad (3.7a)$$
$$a = b - 5 \quad (3.7b)$$

A primeira equação possui o número (`\ref{sub1}`) 3.7a, a segunda (`\ref{sub2}`) 3.7b. Já se quisermos nos referir ao conjunto de equações é só usar (`\ref{eq:sub}`) 3.7.

3.2.9 Linhas

Com o comando `\overline{fórmula}` é criada uma linha acima de uma fórmula e com `\underline{fórmula}` um linha abaixo da fórmula. Também podem ser feitas outros tipos diferentes de linhas usando:

`\overbrace{xyz}` → $\overbrace{xyz}$
`\underbrace{xyz}`→ $\underbrace{xyz}$
`\widetilde{xyz}`→ $\widetilde{xyz}$
`\widehat{xyz}`→ $\widehat{xyz}$
`\overleftarrow{xyz}`→ $\overleftarrow{xyz}$
`\overrightarrow{xyz}`→ $\overrightarrow{xyz}$
`\overline{xyz}`→ $\overline{xyz}$

Há também a possibilidade de se colocar sobre e sobescritos fazendo:

`\overbrace{xyz}^{a}` → $\overbrace{xyz}^{a}$

Ex:

$$\begin{eqnarray*}
\overline{\overline{(\overline{A} \cdot B)} + \overline{(A + \overline{D})}} &=& \\
\overline{\overline{(\overline{A} \cdot B)}} \cdot \overline{\overline{(A + \overline{D})}} &=& \\
(\overline{A} \cdot B) \cdot (A + \overline{D}) &=& \\
(\overline{A} \cdot B \cdot A) \cdot (\overline{A} \cdot \overline{B} \cdot \overline{D}) &=& \overline{A} \cdot B \cdot \overline{D}
\end{eqnarray*}$$

```
\begin{eqnarray*}
\overline{\overline{(\overline{A} \cdot B)} +
\overline{(A + \overline{D})}} &=&  \\
\overline{\overline{(\overline{A} \cdot B)}}
\cdot \overline{\overline{(A + \overline{D})}} &=&  \\
(\overline{A} \cdot B) \cdot (A + \overline{D}) &=&  \\
(\overline{A} \cdot B \cdot A) \cdot
(\overline{A} \cdot \overline{B} \cdot \overline{D}) &=&
\overline{A} \cdot B \cdot \overline{D}
\end{eqnarray*}
```

o comando `\underline` serve para sublinhar e também pode ser usado sem a necessidade de estar em um ambiente matemático.
Ex:`\underline{palavra}` palavra

3.2.10 Empilhando

Usando o comando `\stackrel` é possível colocar um símbolo ou texto acima de outro.
`\stackrel{símbolo acima}{símbolo abaixo}`.

Ex: $_{92}^{239}U^{*} \stackrel{24 min}{\longrightarrow} _{93}^{293}Np^{*} + _{-1}^{0}\beta$

```
$_{92}^{239}U^{*}  \stackrel{24 min}{\longrightarrow}  _{93}^{293}Np^{*} +
_{-1}^{0}\beta $
```

3.2.11 O comando *phantom*

Este comado é bastante útil quando for desejável escrever algo alinhado em sob e sobrescritos.

Ex:

`U_{ij}^{\phantom{i}n}` $\longrightarrow U_{ij}^{n}$

`U_{ij}^{n}` $\longrightarrow U_{ij}^{n}$

Note o alinhamento da letra *n* com as letras *i* e *j*, quando é usado o comando *phantom*.

3.2.12 Espaçamento nas fórmulas

No modo matemático o TeX ignora os espaços dados colocando o espaço que convêm a ele, mas como alguns autores gostam de mudar isso há alguns comandos especiais de espaçamento:

`\,`	pequeno espaço	`\;`	grande espaço
`\:`	médio espaço	`\!`	espaço negativo(backspace)

É bom deixar o TEX colocar o espaço que ele quer, mas como nem tudo é perfeito deve-se ficar atento quando houver símbolos de integral, derivada, raízes e quocientes, pois geralmente o TEX confunde a estrutura lógica.

Ex: ydx é visto como o produto de três variáveis pelo TEX, logo quando digitar isso coloque espaço para que se compreenda que é uma derivada $y\,dx$ → `y\,dx`.

3.2.13 Teoremas

Geralmente um texto matemático possui teoremas, proposições e etc. Para isso o LaTeX tem um comando que define um ambiente com este tipo de estrutura.

Em primeiro lugar através deve ser feita uma declaração com o comando:
`\newtheorem{ambiente}{título}[numeração]`,
onde *ambiente* é um nome qualquer que pode ser dado para posterior uso como o ambiente, *título* é uma denominação que irá aparecer como: teorema, lei, axioma, etc... e *numeração* é a seqüência da numeração que este ambiente irá receber como *chapter*, *section*, é opcional. Este comando pode ser colocado em qualquer parte do seu documento.

Em seguida deve ser usado o ambiente com o nome escolhido para escrever o texto.

```
\begin{ambiente}[nome do teorema]
texto
\end{ambiente}
```

Ex:

```
\newtheorem{ambiente}{Lei}
[chapter]
.
.
.
\begin{ambiente}[Lenz]
O sentido da corrente...
origem.
\end{ambiente}
```

Lei 3.1 (Lenz) *O sentido da corrente induzida é aquele que produz um fluxo magnético que se opõe à variação do fluxo magnético que lhe deu origem.*

A numeração seguirá a mesma seqüência quando for usado novamente o mesmo ambiente.

3.2.14 Tipos especiais de letras

Dentro do ambiente matemático também há a possibilidade de se mudar os tipos de letras da seguinte forma:

`$\mathrm{X Y Z}$` → XYZ
`$\mathcal{X Y Z}$` → $\mathcal{XYZ}$
`$\mathbf{X Y Z}$` → $\mathbf{XYZ}$
`$\mathsf{X Y Z}$` → XYZ
`$\mathtt{X Y Z}$` → $\mathtt{XYZ}$
`$\mathit{X Y Z}$` → XYZ
`$\mathbb{N I Z Q}$` → $\mathbb{NIZQ}$.

Este precisa do pacote *amsfonts*. Para isso basta colocar este comando `\usepackage{amsfonts}` no preâmbulo.

Capítulo 4

Tabelas

Os ambientes *tabbing* e *tabular* são os que permitem alinhar o texto em colunas, mas há algumas diferenças entre eles:

- O ambiente *tabbing* pode ser usado somente no modo de texto; e *Tabular* pode ser usado em qualquer modo (matemático, texto ...).
- O TEX inicia uma nova página no meio do *tabbing* mas não no meio do tabular.
- O TEX determina automaticamente a altura e largura da tabela, enquanto no *tabbing* isso é decretado pelo usuário.

4.1 *Tabbing*

Neste ambiente você alinha as colunas através de marcações inserindo \= para indicá-las. Na linha de baixo, põe-se \> para indicar que está relacionada com a primeira coluna. Já para relacionar com segunda a coluna, põe-se \> duas vezes, ou seja, a linha comecará a partir do segundo marcador \=.

Ex:

```
isto é um exemplo,
        onde começa a próxima
                        palavra
na última linha ?
```

```
\begin{tabbing}
    isto é \= um exemplo, \\
    \> onde começa \= a próxima \\
    \> \> palavra \\
    na última linha ?
\end{tabbing}
```

Note o uso do \>

É importante lembrar que o TEX considera um ou mais espaços como um único espaço, portanto vamos ver um exemplo de erro muito comum cometido neste ambiente:

```
\begin{tabbing}
  coluna         \= curta \\
  coluna grande  \> maior que curta
\end{tabbing}
```

coluna curta
coluna grande maior que curta

Pode ser visto que "maior que curta" ficou por cima de "grande". Por que será, já que foi dado espaço suficiente na linha de cima? Recordando: não importa quantos espaços sejam dados, só será considerado um. Uma dica para evitar que isso ocorra é: colocar a palavra que é maior na primeira linha e elimine-a usando `\kill`. Este comando faz com que a linha seja considerada na compilação, no entanto não apareça na tela.

Ex:

```
\begin{tabbing}
coluna grande \= maior que curta \kill
coluna        \> curta \\
coluna grande \> maior que curta
\end{tabbing}
```

coluna	curta
coluna grande	maior que curta

4.2 *Tabular*

Este é semelhante ao array só diferindo pelo fato de poder ser usado em qualquer modo, não apenas matemático.

Vamos ver um exemplo para entender os comandos.

```
\begin{tabular}{|c|l|rc|}
  \hline
  jan & fev & mar & abr \\ \hline
  mai & jun & jul & ago \\ \cline{1-1} \cline{3-4}
  set & out & nov & dez \\ \hline \hline
\end{tabular}
```

- As letras ***c, l*** e ***r*** significam centro, esquerda e direita respectivamente (center, left e right). Isto indica a posição do texto em relação a célula.
- As barras verticais (|) separando ***c, l*** e ***r*** são para fazer linhas verticais na tabela.
- `\hline` É para fazer linhas horizontais ao longo da tabela.

  Repetindo várias vezes os mesmos comandos (| e `\hline`) seguidamente formam-se várias linhas.
- `\cline{coli-colj}` Faz linhas horizontais apenas entre as colunas i e j.
- `&` Divide os elementos de cada linha.
- `\\` Indica o início de uma nova linha na tabela.

jan	fev	mar	abr
mai	jun	jul	ago
set	out	nov	dez

A largura da coluna pode ser determinada utilizando o comando `p{medida}`, ele deve ser colocado no lugar das letras ***c, l*** ou ***r***

Ex:

col 1	col 2	coluna com 3 cm
col 1	col 2	podemos colocar um frase nesta coluna e ela será quebrada quando o tamanho for maior que 3 cm.
isso não acontece nesta coluna !	col 2	

Há também a possibilidade de se montar uma tabela com multicolunas, ou seja, uma célula grande pode ser construída com o agrupamento de células vizinhas em uma linha. É feito com o comando `\multicolumn{n}{pos}{item}`. Vamos ver o que significa isso na prática.

Primeiro, para a construção de uma tabela, é preciso que se saiba o número máximo de colunas que ela terá para colocá-las no argumento situado após *tabular*. Conforme o comando citado acima, *n* é o número de colunas da tabela inteira que a multicoluna irá ocupar, *pos* é a posição que ficará o texto (*r l c*) e *item* é o texto que será digitado. É o mesmo que construir uma tabela dentro da outra.

Ex:

```
\begin{tabular}{|l|ll|} \hline
segunda & \multicolumn{2}{|c|}{terça}\\ \hline
10 & 15 & 20 \\
15 & 10 & 25 \\ \hline
\end{tabular}
```

segunda	terça	
10	15	20
15	10	25

A tabela é transformada em elemento flutuante, ou seja, ela será colocada pelo LaTeX no local de melhor visualisação quando o ambiente *tabular* é colocado dentro do ambiente *table*.

Para definir o local da página em que a tabela ficará situada pode-se colocar:

h- Ficará onde foi digitado;
b- Ficará na parte inferior da página;
t- Ficará na parte superior da página;
p- Ficará em página separada.

Ex:

```
\begin{table}[b]
\begin{tabular}{}
tabela
\end{tabular}
\end{table}
```

Entretanto, pode ser que o LaTeX não o aceite por motivos estéticos.

4.2.1 Tabelas Longas

Algumas vezes pode haver a nescessidade de se usar grandes tabelas que ocupam mais que uma página, porém o ambiente *tabular* não permite que a tabela seja quebrada de acordo com a página.

O pacote *longtable* permite que isso seja feito através do ambiente *longtable* que deve ser usado no lugar de tabular. Neste caso não é nescessário utilizar o ambiente *table* para tornar

a tabela em um elemnto flutuante. E os comandos do ambiente *longtable* são os mesmos do *tabular*.

Lembre-se que deve-se inserir o pacote no preâmbulo

\usepackage{longtable}

Ex:

```
\begin{longtable}{|c|c|}
 \hline
     &   &    \\
     .
     .
     .
     &   &    \\
  \hline
  \caption{Tabela Longa}
\end{longtable}
```

Capítulo 5

Movendo informações

Neste capítulo será visto como trabalhar com informações no documento através de referência cruzada, bibliografia, citação, inclusão de outros arquivos e figuras, sumário, etc...

5.1 Referência cruzada

Um dos motivos para as figuras, secções, equações e tabelas serem numeradas é para posterior referência delas no texto. Por exemplo, escrevendo diretamente : "consulte a equação 10" pode ocorrer um problema, pois caso seja acrescentada futuramente outra equação antes dessa seu número não será 10, mas sim 11. Logo, a referência estará errada. Para que não aconteça isso pode-se criar um código para aquela equação com o comando `\label{código}` e referenciá-la com o comando `\ref{código}`.

Ex:

```
\begin{equation}
a^{2}+ b^{2}=c^{2} \label{Eq:pitágoras}
\end{equation}

Consulte a \ref{Eq:pitágoras} que é a equação de Pitágoras.
```

$$a^2 + b^2 = c^2 \tag{5.1}$$

Consulte a 5.1 que é a equação de Pitágoras.

Conforme for sendo acrescentada mais referências será preciso rodar o LaTeX mais de uma vez para que ele atualize a lista.

Também pode ser usado `\pagref{...}` para referência de páginas. E para referênciar partes do documento como capítulos e secções basta fazer o mesmo que é feito com a equação. Ex:

`\section{Referência cruzada \label{sec:rc}}`.

Uma dica para não se perder com a quantidade de códigos diferentes é na hora de colocar referência identificá-la com: eq:(equação), fig:(figura), tab:(tabela), teo:(teorema)....

É preciso compilar o LaTeX duas vezes, na primeira ele guarda a informação em um arquivo auxiliar e na segunda ele vai até este arquivo buscar esta informação e colocar no documento.

5.2 Dividindo o arquivo

Quando o arquivo fica muito grande é importante que ele seja dividido em arquivos menores, para que o tempo de processamento seja menor,e que no final seja reunido novamente. Isso é feito da seguinte forma:

```
\includeonly{lista de arquivos separados por
 vígula sem .tex no preâmbulo}
\include{arquivo sem .tex na ordem desejada
dentro do ambiente \textit{document}}
```

Ex:

```
\includeonly{introducao,formatacao,capa}
. . .
\include{capa} ...
%\include{formatacao} ...
%\include{introducao} ...
```

Usando `\input` no lugar de `\include`, o arquivo é inserido no meio da página onde foi colocado, ao contrário do `\include` que inicia uma nova página.

O LaTeX faz automaticamente o ajuste de numeração na hora em que reúne os arquivos.

Obs: Os arquivos que serão inseridos no documento não devem ter os comandos que aparecem no preâmbulo, nem `\begin` e `\end{document}`.

Outra vantagem de separar os arquivos é que a compilação pode ser feita separadamente para cada parte do documento, para isso basta colocar o símbolo de comentário, %, como feito no exemplo acima, e só será compilado aquela parte que está sem o símbolo. Isto facilita a localização de erros.

5.3 Bibliografia

Uma das maneiras de fazer a bibliografia é utilizar o ambiente : *thebibliography*. E cada referência começa com `\bibitem{livro}` e o comando `\cite{livro}` faz as referências no meio do documento.

```
\begin{thebibliography}{n}
    \bibitem{ref}{Livro}
\end{thebibliography}
```

Onde *n* é o número máximo de itens de referência que terá o documento, *ref* é o código de referência do livro e *livro* são os dados do livro. Este ambiente deve ser colocado no final do documento.

BibTeX

A outra maneira é através do utilitário BibTeX.

Ele permite que sejam montados dados bibliográficos para posterior uso em seus documentos.

É usado \cite para citações no meio do texto, mas em vez de digitar a lista diretamente no documento, é usado o comando \bibliography{nome} com o nome dos arquivos que contém o banco de dados.

Antes de saber com fazer isso, vamos ver como o LaTeX e o BibTeX interagem. Quando o documento é compilado pelo *latex* é criado um arquivo com extensão .aux que contém todas as informações de referência cruzada. Quando seu documento tiver os comandos \bibliography e \bibliographystyle, este arquivo *.aux guardará as informações de citações e argumentos deste comando. Em seguida quando o documento é compilado pelo *bibtex*, todas essas imformações são lidas e é criado um novo arquivo com extensão .bbl contendo os comandos que produziram a lista. A próxima vez em que o documento for compilado pelo *latex* o comando \bibliography lê o arquivo *.bbl e gera a lista bibliográfica.

Como fazer:

- Primeiramente deve-se escrever um arquivo (é importante ressaltar que este é um arquivo separado do documento principal) contendo os dados bibliográficos baseado nos tipos pré-definidos pelo LaTeX. Os principais são:

  article São os artigos de jornais ou revistas.

  book Um livro.

  inbook Parte de um livro (capítulo, páginas, etc...).

  manual Documento técnico.

  Os principais campos que devem ser preenchidos são:

  author Autor
  title Título
  year Ano
  publisher Editora
  address Endereço (cidade, estado...).

  Consulte a referência [1] para encontrar mais tipos.

```
Tipo@{código de referência,
        AUTHOR= "nome",
        TITLE = " Título ",
        PUBLISHER = "Editora",
        ADDRESS = "Endereço",
        . . .
        YEAR = "Ano" }
```

  Geralmente as interfaces gráficas disponíveis para o trabalho com o LaTeX, já dispolibilizam opções em uma forma completa, cabendo ao usuário somente o preenchimento dos campos. Consulte o manual do seu editor.

  Salve este arquivo com extensão .bib na mesma pasta onde está o documento;

- Insira o comando `\bibliographystyle{estilo}` no preâmbulo do documento e o comando `\bibliography{arquivo sem .bib}` no local onde deverá apareçer a bibliografia.

  Os estilos podem ser:

  plain É o mais usado. As entradas são colocadas em ordem alfabética.

  unsrt As entradas aparecem na ordem de citação no meio do texto.

  abbrv Semelhantes aos anteriores, mas vem com nomes abreviados.

  Obs: Só apareçerão na lista bibligráfica as referências que foram citadas, para que elas apareçam na lista, sem precisar de serem citadas no texto use o comando:

  `\nocite{nome dos códigos separados por vírgula}`.

- Para gerar a bibliografia deve-se compilar o *latex*, em seguida o *bibtex* e o *latex* novamente.

Por exemplo:
É criado um arquivo contendo a descriçao da referência, como feito abaixo:

```
@book{ apos:tex,

author = {Programa Especial de Treinamento},

title ={Apostila de \LaTeX},

publisher = {Universidade Federal Fluminense},

address = {Niterói - Rio de Janeiro,

year ={2004}
}
```

O arquivo é então salvo na mesma pasta com um nome qualquer e extensão .bib. Ex: livros.bib

É inserido no preâmbulo o documento o comando: `\bibliographystyle{estilo}`. E no fim do documento o comando: `\bibliography{arquivo sem .bib}`.

Ex:

```
\bibliographystyle{plain}
.
\begin{document}
.
.
\bibliography{livros}
\end{document}
```

Feito isso, é só rodar o *latex* - *bibtex* -

5.4 Figuras

Para inserir figuras deve-se colocar no preâmbulo o pacote *graphicx* e depois usar o comando que permite inserir figura. Inicialmente, a extensão de figura que o LaTeX aceitava era somente: *EPS* (*Encapsulated PostScript*), por ser mais usada. Porém, a nescessidade de inclusão de figuras com extensões diferentes era grande, então, alguns usuários do LaTeX criaram pacotes que permitiam a inserção de outros formatos de figuras além do .eps. É usado o ambiente *figure* para poder tornar a figura um elemento flutuante, dar um título à figura e usar um código para referência.

```
\begin{figure}[argumento de posição como no table]
\includegraphics[medidas]{nome do arquivo}\\
\caption{título da figura}\label{código de referência}
\end{figure}
```

As *medidas* são os parâmetros:

- *width* Largura;
- *height* Altura;
- *angle* Rotaciona a figura no sentido horário.
- *scale* Muda a escala da figura.

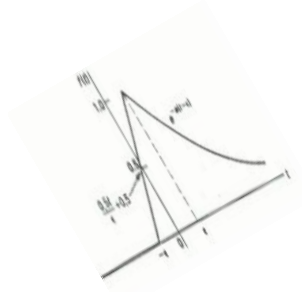



Figura 5.1: Gráfico

```
\usepackage{graphicx}
...
\begin{document}
...
\begin{figure}[h]
\centering % este comando é usado para centralizar a figura
\includegraphics[width=4cm, height=6cm, angle=30]{grafico.jpg}\\
\caption{Gráfico}\label{fig:exem}
\end{figure}
```

Figuras .pdf Se o pacote *graphicx* for usado com o opcional [*pdftex*] fica possível inserir figuras no formato *.pdf, neste caso o documento não poderá ser compilado com o *latex* e sim com o *pdflatex*. Deve-se conferir se seu sistema oferece este recurso.

Ex:

```
\usepackage[pdftex]{graphicx}
...
\begin{document}
...
\begin{figure}[h]
\centering
\includegraphics{grafico.pdf}\\
\caption{Gráfico}\label{fig:exem}
\end{figure}
```

Compilando com o *pdftlatex* é gerado diretamente o documento no formato .pdf ao invés de .dvi.

Figuras .jpg, .png, .pdf Para inserir figuras neste formato deve-se em primeiro lugar inserir comando *dvipdfm* como argumento opcional da classe do documento.
`\documentclass[dvipdfm]{report}`

Para inserir a figura é necessário transformá-la do formato original (.jpg, .png, .pdf) para o formato .bb (*bounding box*). Como fazer?

Suponha que o nome da arquivo seja `grafico.jpg`. Para transformá-lo em `gráfico.bb` basta abrir o *Command Prompt* e digitar:

`ebb grafico.jpg`

Isto transforma o arquivo de `.jpg` em `.bb`.

Obs: A figura não aparecerá no documento `.dvi`. É preciso que se transforme-o em .pdf. Para fazer isso, abra o *Command Prompt* e digite:

`dvipdfm documento.dvi`

Supondo que o nome do arquivo seja *documento*.

5.4.1 Subfiguras

Existe um pacote chamado *subfigure* que permite o ambiente *figure* possuir mais que uma figura.

Antes de qualquer coisa deve-se declarar no preâmbuo o comando:

`\usepackage{subfigure}`

Veja o exemplo abaixo para entender o uso do comando.

Ex:

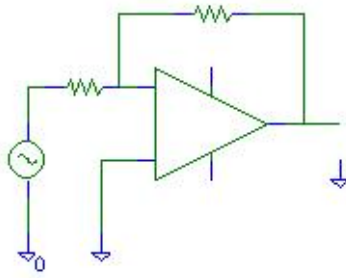
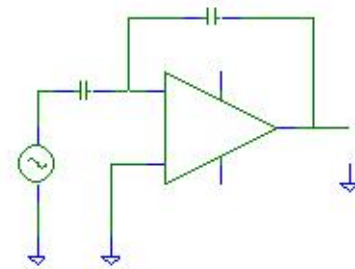
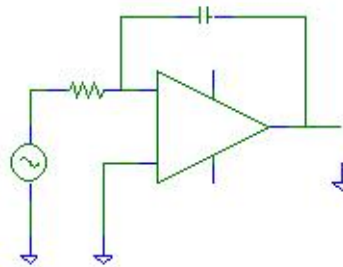
```
\begin{figure}[h]
\subfigure[Primeira\label{fig:pri}]{\includegraphics{fig1.jpg}}
\subfigure[Segunda\label{fig:seg}]{\includegraphics{fig2.jpg}}
\center{\subfigure[Terceira\label{fig:ter}]{\includegraphics{fig3.jpg}}}
\caption{Conjunto de figuras}\label{fig:conj}
\end{figure}
```

(a) Primeira (b) Segunda

(c) Terceira

Figura 5.2: Conjunto de figuras

O título de cada uma das figuras 5.2 `\ref{fig:conj}` é colocado entre colchetes, assim como seu código de referência. A terceira figura 5.2(c) `\ref{fig:ter}` esta dentro do comando `\center` para ficar centralizada.

5.5 Sumário

O sumário é feito, facilmente, através de um único comando: `\tableofcontents`, que deve ser colocado logo após `\begin{document}`, e o sumário é gerado automaticamente.

Ex:

```
...
\begin{document}
\tableofcontents

....
```

Caso haja alguma secção, figura ou tabela sem a numeração (por exemplo, quando é colocado o nome da secção com asterisco: `\section*`. Veja: 2.2.13) é possível que ela apareça no sumário usando o comando
`\addcontntsline{arquivo}{secção}{nome}`, onde *arquivo* é a extensão da lista que deverá entrar (veja 5.7), *secção* é o título secção, capítulo e *nome* é o nome que apareçerá na lista, pode ser o mesmo título.

Ex:

```
\tableofcontents
...
\addcontentsline{toc}{chapter}{Prefácio}
```

Os comandos \listoffigures e \listoftables geram um lista de figuras e lista de tabelas, respectivamente.

É preciso compilar o arquivo duas vezes para que seja visualizado o sumário a cada mudança.

5.6 Índice

Uma das maneiras de produzir o indice remissivo é utilizando o ambiente *theindex* onde cada entrada é feita usando \item, a subentrada usando \subitem e a subsubentrada usando \subsubitem. Este ambiente produz o índice em duas colunas. O comando \indexspace faz um espaço vertical entre as entradas.

Ex:

```
\item babel 14

\item Color 44

    \subitem \verb=\=textcolor 56, 32

        \subsubitem color 45
\indexspace

\item article 15
```

Faz

```
babel 14
Color 44
    \textcolor 56, 32
        color 45

article 15
```

O índice remissivo também pode ser criado facilmente da seguinte forma:

O programa *MakeIndex* Este é um programa que faz índice em um documento gerado pelo LaTeX.

Primeiro coloque no preâmbulo o pacote *makeidx* e o comando \makeindex. E no local onde deverá aparecer o índice ponha o comando: \printindex.

Para marcar os itens que apareceram no índice use o comando: \index{item}, onde *item* é a palavra que aparecerá no índice (*entrada*). Isso faz aparecer o item ao lado da página onde ele está localizado. *Subentradas* também podem ser geradas da seguinte forma: \index{item!item!item}. O ponto de exclamação serve para separar as entradas das subentradas.

Ex:

```
\documentclass[a4,12pt,oneside]{article}
\usepackage[T1]{fontenc}
\usepackage[brazil]{babel}
\usepackage{makeidx}
\makeindex
\begin{document}
...

\index{babel}
 ...
\index{Color!\verb=\=textcolor!color}
 ...
\index{Classe!book!article}

\printindex
\end{document}
```

Usando o exemplo acima, na impressão deverá aparecer no índice remissivo o seguinte:

```
babel, 12

Classe
   book
      report, 7

Color
   \textcolor
     color, 30
```

Para gerar o índice é preciso:

- Compilar o documento pelo *latex* para que ele gere um arquivo com extensão .idx;
- Em seguida compilar pelo *makeindex* com o nome do arquivo com extensão .idx para que ele gere um novo arquivo com extensão .ind;
- E compilar pelo *latex* novamente.

  Sempre que houver uma mudança deve-se fazer esta etapa de compilação.

5.7 Tipos de arquivos

Quando o documento é compilado, vários arquivos serão criados com o mesmo nome do documento. A lista abaixo dá uma breve explicação do significado de cada extensão de arquivo.

.tex Este é o arquivo principal, onde está o código fonte escrito.

.dvi Este é o arquivo pronto para ser visualizado e transformado em *ps* para imprimir. É o arquivo independente de dispositivo (*device independent*).

.aux É onde está localizada as informações de referência cruzada.

.toc Armazena os títulos das secções *(Table of contents)*.

.lof Armazena os títulos das figuras *(List of figures)*.

.lot Armazena os títulos das tabelas *(List of tables)*.

.idx Contém o índice remissivo.

.log É o relatório da compilação feita, com os erros.

.bbl É o arquivo escrito pelo BibTex.

Capítulo 6

Estrutura visual

Neste capítulo será visto conceitos básicos sobre como modificar a formatação padrão .

6.1 Cabeçalho e Rodapé da página

A página é formada por 3 partes: cabeçalho (topo da página), corpo (o texto) e rodapé (parte inferior da página). As características do cabeçalho e do rodapé da página são definidas pelo comando `\pagestyle`, enquanto que a numeração é definida com o comando `\pagenumbering`.

Os quatro estilos principais da página são feitos usando o comando `\pagestyle{estilos}`, onde os estilos são:

plain A página possui apenas numeração no rodapé;

empty Produz cabeçalho e rodapé vazio , sem numeração;

headings Depende do estilo do documento.
A página da frente (ímpar) vem com a secção no caso de *book* e subsecção no caso de *article*. Quando a página estiver definida como *oneside* aparecerá nas páginas, o capítulo no caso de *book* e secção no caso de *article*. Caso não haja secções só aparece a numeração no cabeçalho;

myheadings É especificado pelo usuário o que estará escrito no topo com os comandos:

- markboth – Quando o documento estiver em *twoside* `\markboth{página par}{página ímpar}`;
- markright – Quando o documento estiver em *oneside* `\markright{páginas pares e ímpares}`.

  Nos locais onde estão escritos página par e/ou ímpar é que deveram ser escritas as palavras que aparecerão no cabeçalho.

A numeração da página é feita automaticamente com algarismos árabes, mas, para mudá-los, use o comando:
`\pagenumbering{roman}` Os números são colocados em romano;

Roman em romano maiúsculo;
alph em letras comuns e

Alph em letras comuns maiúscula.

Se \pagenumbering for colocado no meio do texto, a numeração a partir daí será iniciada novamente com o novo estilo de numeração declarado.

O pacote *fancyhdr* Os comandos que foram vistos acima são úteis, porém limitados. Um pacote que adiciona recursos ao estilo da página é o *fancyhdr*.

Com ele pode-se, entre outras coisas:

- Dividir o rodapé e cabeçalho em 3 partes diferentes, e
- Inserir linhas;

Em primero lugar deve-se inserir o pacote através do comando: \usepackage{fancyhdr} no preâmbulo, no caso do LaTeX2e. E no caso do LaTeX2.09 insira fancyhdr como um argumento opcional da classe do documento: \documentstyle[fncyhdr,12pt,a4]{book}

Após isso defina no preâmbulo o estilo da página através do comando \pagestyle{fancy}.

Para páginas ímpares (ODD)

RO esquerda	CO (cabeçalho)centro	LO direita
	corpo da página	
RO esquerda	CO (rodapé) centro	LO direita

Para páginas pares (EVEN)

RE esquerda	CE (cabeçalho)centro	LE direita
	corpo da página	
RE esquerda	CE (rodapé) centro	LE direita

Cada campo é definido pelos seguintes comandos:

- \fancyhead[parâmetro]{informação}
- \fancyfoot[parâmetro]{informação}

O usuário pode definir como *informação* o que quiser, ou colocar também o comando \thepage que diz onde o número da página vai aparecer.

O parâmetro é a indicação de em qual campo a informação irá aparecer. Por exemplo, observe a tabela 6.1. As letras O e E dizem respeito às páginas ímpares e pares, respectivamente. C, R e L, significam centro, direita e esquerda, respectivamente. Através de cada combinação dessas letras é que fica especificado onde a imformação estará.

Utilizando os comandos:
\renwcommand{\headrulewidth}{medida} e \renwcommand{\footrulewidth}{medida}

Para páginas ímpares (ODD)

Capítulo	Secção
corpo da página	
Número da página	

Para páginas pares (EVEN)

Secção	Capítulo
corpo da página	
Número da página	

é possível estabelecer a largura das linhas do cabeçalho e rodapé, respectivamente, através da medida dada.

Caso não seja usado nehum desse comandos, o padrão que será gerado será:

Este padrão é produzido da seguinte forma:

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
```

Repare que `\rightmark` está se referindo à secção e `\leftmark`, ao capítulo. Caso o usuário queira, estes comandos também podem servir como *informação*.

O exemplo abaixo mostra como foi feito o cabeçalho desta apostila:

```
\pagestyle{fancy}
\fancyhead[LO,LE]{\textit\nouppercase\leftmark}
\fancyhead[RO,RE]{\textit\nouppercase\rightmark}
\fancyfoot[LO,LE]{\textsc{\uff}}
\fancyfoot[RO,RE]{\pet}
\fancyfoot[CO,CE]{\thepage}
\renewcommand{\footrulewidth}{0.4pt}
\renewcommand{\headrulewidth}{0.4pt}
```

O comando `\nouppercase` utilizado acima tem a função de colocar os capítulos e secções em letras minúsculas.

6.2 Área de impressão

A página em qualquer documento em LaTeX possui uma determinada configuração que depende de vários paramêtros, isto é, comandos que agem em determinadas parte da página. A figura 6.1 mostra todos esses comandos.

Cada tipo de classe de documento e papel escolhido tem um tamanho padrão para a impressão. Mas, é possível mudar esse tamanho. Uma das maneiras é através dos comandos:

`\addtolength{padrão}{medida}` – Ele adicionará o medida que escolhida ao padrão de medida do documento, e;

`\setwidth{padrão}{medida}` – Fixa a medida escolhida para o padrão. O comando anterior adiciona uma medida enquanto esse dá uma nova medida.

padrão são os comandos:

- `\textwidth` A largura padrão do texto na página;
- `\textheight` A altura padrão do texto na página;
- `\columnsep` A largura do espaço entre as colunas quando *twocolumn* estiver como opção de estilo;
- `\columnseprule` A largura da linha vertical colocada entre as colunas do texto usando *twocolumn*. O padrão é largura zero, por isso não aparece a linha normalmente;
- `\oddsidemargin` É a distância da borda esquerda do papel para a margem esquerda do texto menos uma polegada, em páginas ímpares quando *twoside* estiver declarado;
- `\evensidemargin` É o mesmo que `\oddsidemargin` só que para páginas pares;
- `\marginparwidth` Largura das notas marginais;
- `\topmargin` A distância da margem superior do papel para o topo do cabeçalho da página menos uma polegada;
- `\headheight` A altura da caixa que contém o cabeçalho.

E a *medida* pode ser colocada em centímetros ou outra medida citada na secção 6.3.

Ex: `\addtolength{\textheight}{3cm}` adicionará 3 centímetros à altura área de impressão da página

Este comando deve ser colocado no preâmbulo.

6.3 Espaços e Medidas

Através dos seguintes comandos pode ser dado espaçamento entre linhas e palavras:

- `\hspace{medida}` - Põe espaço entre as palavras, onde a medida pode ser em:
  - (cm) Centímentros;
  - (*in*) Polegada $(1in = 2.54cm)$;
  - (*pt*) Ponto $(1in = 72.27pt)$;
  - (*ex*) Medida equivalente a altura da letra *x*;
  - (*em*) Medida relativa a largura da letra *M*.

  É melhor que se use as duas últimas medidas por elas serem baseadas na fonte usada no documento.

  Ex:

  ```
  Este espaço\hspace{10em} tem 10em
  ```

  Este espaço   tem 10em.

Figura 6.1: Medidas da página

- `\vspace{medida}` - Põe espaço vertical entre as linhas.
- `\hrulefill` -Produz uma linha horizontal.
  Ex:
  `Linha \hrulefill horizontal .`
  Linha ____horizontal.
- `\rule[elevação]{largura}{altura}` – Faz um retângulo com as medidas *elevação* acima ou abaixo da linha , *largura* e *altura*

  Ex:
  texto `\rule[0.5ex]{5em}{0.7ex}` texto

  texto ▬▬▬ texto
- `\dotfill` Produz linha horizontal pontilhada.
  Ex:
  `linha \dotfill pontilhada.`
  linha .... pontilhada.
- `\hfill` Produz um espaço em branco com tamanho proporcional ao número de palavras na linha.
- `\vfill` Adiciona espaçamento vertical na página proporcionalmente ao número de linhas que ela possui.

  O tamanho da linha nos comandos *fill* são ajustados de acordo com a quantidade de palavras na linha ou página. Linha ________________horizontal menor que a anterior.

Obs: Outros padrões de medida de espaço utilizado pelo LaTeX e podem servir de base para outras medidas de outros comandos são:

- `\parindent` É o tamanho da indentação no parágrafo normal.
- `\parskip` É o espaço vertical entre os parágrafos.
- `\baselineskip` É a distância entre o topo de uma linha e o topo da linha de baixo.
- `\linewidth` É igual ao comando ceto quando estiver em ambientes como quote e os de lista onde ele define a largura destes ambientes. Seu valor não pode ser mudado, mas serve com padrão para outras medidas.

  Esses três primeiros padrões podem ser mudadas no seu documento colocando no preâmbulo o comando:
  `\addtolength{padrão}{medida}`, onde *padrão* é a medida que será mudada e *medida* é o novo espaço.

6.4 Caixas

Usando o comando `\fbox{palavra}` É construída uma caixa ao redor da palavra. Exemplo.
E através do comando `\framebox[medida][posição]{palavra}` pode-se controlar o tamanho da caixa.
Ex: texto `\framebox[6ex][r]{palavra}` texto.

texto palavra texto.

Usando o comando `\makebox` da mesma forma que `\framebox` também é criada uma caixa, só que sem linha.
Ex: texto `\makebox[6ex][r]{palavra}` texto.

texto palavra texto.

O comando `\raisebox{altura}{palavra}` faz uma caixa onde a *palavra* é elevada uma *altura* escolhida, podendo também ser uma medida negativa (para baixo -).
Ex: texto `\raisebox{1.0ex}{palavra}` texto.

texto palavra texto.

6.5 Cores

Primeiramente, para usar cor é preciso que se ponha no preâmbulo o comando `\usepackage{color}`. Isso é para o LaTeX carregar o pacote *color* e reconhecer os comandos de cores. Veja alguns comandos:

- `\definecolor{nome}{modelo}{parâmetro}` - Este define a cor, onde *nome* é o nome da cor, *modelo* é o modelo da cor com o principaL sendo: *rgb* (red, green, blue) e *parâmetro* é o código da cor segundo o modelo usado. Ex: `\definecolor{azul}{rgb}{0,0,1}` define a cor azul.

  O padrão *rgb* é o mais usado devido ao seu grande número de combinação de cores feita com os códigos: peso da cor vermelha, peso da cor verde, peso da cor azul, onde os números variam entre 0 e 1.

- `\textcolor{cor}{palavra}` - Muda cor da palavra selecionada. A cor pode ser escrita diretamente em inglês ou usar o *definecolor* definindo o nome da cor em português.

  Ex:
  ```
  \definecolor{verdemar}{rgb}{0.59,0.78,0.65}
  \textcolor{verdemar}{texto em cor personalizada}
  ```
  texto em cor personalizada

- `\color{cor}` - Muda a cor do texto inteiro. Para mudar apenas um trecho do texto use chaves neste trecho.
  Ex:
  ```
  {\color{blue}{trecho do texto}}
  ```
  trecho do texto

- `\pagecolor{cor}` - Muda a cor do fundo da página.

- `\colorbox{cor}{texto}` Gera uma caixa com o fundo da cor que foi escolhida. Para fazer uma caixa em um grande pedaço de texto use *minipage*, veja secção 6.6.
  Ex:
  `\colorbox{red}{palavra}`
  palavra
- `\fcolorbox{bcor}{ccor}{texto}` - Gera uma caixa com *ccor* e borda *bcor*.
  Ex:
  `\fcolorbox{verdemar}{green}{palavra}`
  palavra

Importante: dependendo do visualizador de DVI as cores não serão vistas, mas se converter em PS ou PDF , apareçerá normalmente.

6.6 Minipage

Esse é um ambiente que cria uma área com formato de uma página, com largura desejada através do comando:

```
\begin{minipage}[posição t ou b]{largura}
   texto
\end{minipage}
```

O texto pode conter outros ambientes, podendo colocar até notas de pé de página [a], mas esta nota não apareçerá no fim da página comum. Há também possibilidade de se colocar moldura com o comando `\fbox{ambiente minipage}` e também trabalhar com os comandos de cores sem problema, através de combinações. Os argumentos *b* e *t* permitem o alinhamento do topo (t) e do fim (b) da *minipage* em relação a linha do texto, deve ser usado quando há uma outra *minipage* do lado.

[a] esta nota aparece no fim da minipágina

```
\begin{minipage}[t]{0.5\textwidth}
O texto pode conter ... de página ...
 \footnote{esta nota
...ipágina}, ... do lado.
\end{minipage}
```

Outra posibilidade de colocar moldura é usando *tabular*, onde os itens serão as minipáginas.

6.7 Novos comandos e ambientes

O LATEX também permite que seja modificado o nome de algum comando para o nome escolhido ou crie uma macro, ou seja, um comando que sintetize outros comandos. Isso é feito através de `\newcommand{novo comando}{definição}`

Ex: Se o texto deve ter escrito a mesma frase várias vezes ao longo dele, seria útil usar algo que resumisse isso. Então, querendo digitar: Universidade Federal Fluminense, coloca-se no preâmbulo
`\newcommand{\uff}{Universidade Federal Fluminense}`. Depois é só usar `\uff` para aparacer a frase: Universidade Federal Fluminense.

Pode-se também montar um comando que tenha uma estrutura em que os argumentos variem. Ex:

```
\newcommand{\vt}[3]{\emph{vetor}$(#1;#2;#3)$}
os vetores \vt{5x}{3x}{7x} e \vt{9w}{8w}{3w} são ...
```

Faz

os vetores *vetor*$(5x;3x;7x)$ e *vetor*$(9w;8w;3w)$ são ...

Vamos ver o que significa cada coisa:

- `\vt` É o nome dado ao novo comando.
- O [3] É o numero de argumentos que variam. No caso as 3 coordenadas.
- `#1;#2;#3` Indica o local em que aparecerá os argumentos.
- `\vt{}...` é o uso do comando onde em cada chaves são colocados os argumentos.

Para fazer ambientes há uma pequena diferença:

```
\newenvironment{emphit}{\begin{itemize}\em}{\end{itemize}}
\begin{emphit}
\item este é o novo item enfatizado
\end{emphit}
```

Faz:

- *este é o novo item enfatizado*

O que foi feito:

- emphit É o nome do novo ambiente.
- *itemize* É o ambiente base pois geralmente os novos ambientes são feitos a partir de um existente.
- `\emph` Faz o texto do item ficar enfatizado.

Também podem ser definidos ambientes com argumentos variaveis, assim como comandos. Ex: No preâmbulo colocando:

```
\newenvironment{meuambi}[1]
{\begin{center}
\fbox{\rule{1ex}{1ex}\hspace{15ex}{#1}\hspace{15ex}
\rule{1ex}{1ex}}}
 {\end{center}}
```

E no meio do texto:

```
\begin{meuambi}
{Exemplo}
\end{meuambi}
```

Será visto o seguinte ambiente:

■ Exemplo ■

Apêndice A

Utilizando o LaTeX através de um Terminal de Comando

Geralmente, os usuários costumam utilizar o sistema LaTeX através de editores de textos específicos e que já possuem em sua interface os comandos de compilação através de ícones, bastanto clicá-los para que se tenha os arquivos gerados em `.dvi, .pdf`, etc... Porém, há casos em que é nescessário utilizar o LaTeX diretamente através de um terminal de comando, principalmente para os usuários de Linux, que possui ainda poucas opções de bons editores.

Este capítulo tem a intenção de mostrar ao usuário alguns comandos básicos para o completo trabalho com o LaTeX.

Suponha que o arquivo principal chama-se: `arquivo.tex` e que o arquivo que contenha a lista bibliográfica se chame: `refer.bib` .

DVI Para comilar o arquivo e gerar um documento em .dvi, digite:

```
latex arquivo.tex
```

PS Para transformar o .dvi em .ps, digite:

```
dvips arquivo.dvi
```

PDF Para transformar o .dvi em .pdf, digite:

```
dvi2pdf arquivo.dvi
```

Para gerar transformar o documento de .tex diretamente para .pdf, compile o arquivo digitando:

```
pdflatex arquivo.tex
```

BIBTEX Para gerar o arquivo em .dvi contendo a lista bibliográfica do arquivo .bib, digite:

```
latex arquivo.tex

bibitex refer.tex

latex arquivo.tex
```

MAKEINDEX Para gerar o arquivo em .dvi contendo indíce remissivo utilizando o pacote *makeidx* , digite:

```
latex arquivo.tex

makeindex arquivo.idx

latex arquivo.tex
```

Apêndice B

Símbolos matemáticos

Todos esse símbolos são usados apenas em ambientes matemáticos, portanto para inserí-los no meio de um texto use $ ... $.

$\lim$	`\lim`	$\arg$	`\arg`	$\cot$	`\cot`
$\liminf$	`\liminf`	$\ker$	`\cos`	$\coth$	`\coth`
$\arccos$	`\arccos`	$\lg$	`\lg`	$\max$	`\max`
$\arcsin$	`\arcsin`	$\cosh$	`\cosh`	$\csc$	`\csc`
$\arctan$	`\arctan`	$\ln$	`\ln`	$\min$	`\min`
$\det$	`\det`	$\exp$	`\exp`	$\hom$	`\hom`
$\sec$	`\sec`	$\sinh$	`\sinh`	$\tan$	`\tan`
$\dim$	`\dim`	$\gcd$	`\gcd`	$\inf$	`\inf`
$\sin$	`\sin`	$\sup$	`\sup`	$\tanh$	`\tanh`

Tabela B.1: Funções

$\hat{}$	`\hat{}`	$\check{}$	`\check{}`	$\breve{}$	`\breve{}`	$\acute{}$	`\acute{}`
$\grave{}$	`\grave{}`	$\tilde{}$	`\tilde{}`	$\bar{}$	`\bar{}`	$\vec{}$	`\vec{}`
$\dot{}$	`\dot{}`	$\ddot{}$	`\ddot{}`				

Tabela B.2: Acentos matemáticos

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons				

Tabela B.3: Setas

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\models$	\models
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim	$\perp$	\perp
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq	$\mid$	\mid
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp	$\parallel$	\parallel
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\smile$	\smile
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\neq$	\neq	$\frown$	\frown
$\in$	\in	$\ni$	\ni	$\doteq$	\doteq		
$\vdash$	\vdash	$\dashv$	\dashv	$\propto$	\propto		

Tabela B.4: Símbolos de relação

$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\bigtriangleup$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$\ast$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\vartriangleleft$	\vartriangleleft	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\vartriangleright$	\vartriangleright	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\trianglelefteq$	\trianglelefteq	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\trianglerighteq$	\trianglerighteq	$\amalg$	\amalg

Tabela B.5: Símbolos de operação binária

Minúscula

α	\alpha	θ	\theta	o	o	τ	\tau
β	\beta	ϑ	\vartheta	π	\pi	υ	\upsilon
γ	\gamma	ι	\iota	ϖ	\varpi	ϕ	\phi
δ	\delta	κ	\kappa	ρ	\rho	φ	\varphi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	χ	\chi
ε	\varepsilon	μ	\mu	σ	\sigma	ψ	\psi
ζ	\zeta	ν	\nu	ς	\varsigma	ω	\omega
η	\eta	ξ	\xi				

Maiúscula

Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Tabela B.6: Letras Gregas

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall	∞	\infty
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists	$\Box$	\Box
$\imath$	\imath	∇	\nabla	$\neg$	\neg	$\Diamond$	\Diamond
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat	$\triangle$	\triangle
ℓ	\ell	$\top$	\top	$\natural$	\natural	$\clubsuit$	\clubsuit
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp	$\diamondsuit$	\diamondsuit
$\Re$	\Re	$\Vert$	\\|	$\backslash$	\backslash	$\heartsuit$	\heartsuit
$\Im$	\Im	$\angle$	\angle	∂	\partial	$\spadesuit$	\spadesuit
$\mho$	\mho	$\sum$	\sum	$\prod$	\prod	$\coprod$	\coprod
$\int$	\int	$\oint$	\oint	$\bigcap$	\bigcap	$\bigcup$	\bigcup
$\bigsqcup$	\bigsqcup	$\bigvee$	\bigvee	$\bigwedge$	\bigwedge	$\bigodot$	\bigodot
$\bigotimes$	\bigotimes	$\bigoplus$	\bigoplus	$\biguplus$	\biguplus	$\dots$	\dots
$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots		

Tabela B.7: Símbolos variados

Apêndice C

Outros símbolos

Acentos

ò	\`{o}	õ	\~{o}	ǒ	\v{o}	¸	\c{}
ó	\'{o}	ō	\o=	ő	\H{o}	.	\d{}
ô	\^{o}	ȯ	\.{o}	o͡	\t{o}	_	\b{}
ö	\"{o}	ŏ	\u{o}				

Símbolos estrangeiros

œ	\oe	å	\aa	ł	\l	¿	?`
Œ	\OE	Å	\AA	Ł	\L	¡	!`
æ	\ae	ø	\o	ß	\ss		
Æ	\AE	Ø	\O				

Tabela C.1: Símbolos estrangeiros e acentos

†	\dag	¶	\P	✓	\checkmark	¥	\yen
‡	\ddag	©	\copyright	✠	\maltese	&	\&
§	\S	£	\pounds	®	\circledR	%	\%
#	\#	_	_	$	\$	LaTeX	\LaTeX
{	\{	}	\}	LaTeX 2$_\varepsilon$	\LaTeXe	TeX	\TeX

Tabela C.2: Símbolos diversos

Referências Bibliográficas

[1] Lamport, Leslie, LaTeX: *A Document Preparation System, User's Guide & Reference Manual*, Addison-Wesley Plubshing Company, 1986.

[2] T. Oitker, H.Partl, I. Hyna, E.Schlegl, *The Not so short introduction to LaTeX 2_ε*, Tradução D. A. Polli, USP, 2000.

[3] H. Kopka e P. W. Daly, *A guide to LaTeX 2_ε, Document preparation for beginners and advanced users*, Adisson-Wesley Plubshing Company, 1995.

[4] **CTAN** (Comprehensive TeX Archive Network), *www.ctan.org*, Este site é referência mundial para materiais relacionados ao LaTeX.

[5] **TeX-Br**, *http://biquinho.furg.br/tex-br/*, Página dos usuários brasileiros de LaTeX.

[6] **Site-LaTeX**, *www.vila.bol.sitelatex.com.br*, Contém vários *links* relacionados ao LaTeX.z

Índice Remissivo